Wind power predictions from nowcasts to 4-hour forecasts: a learning approach with variable selection

Dimitri Bouche[a] [*] , Rémi Flamary[d] , Florence d'Alché-Buc[a] , Riwal Plougonven[b] , Marianne Clausel[c] , Jordi Badosa[b] , Philippe Drobinski[b]

Abstract

We study the prediction of short term wind speed and wind power (every 10 minutes up to 4 hours ahead). Accurate forecasts for those quantities are crucial to mitigate the negative effects of wind farms' intermittent production on energy systems and markets. For those time scales, outputs of numerical weather prediction models are usually overlooked even though they should provide valuable information on higher scales dynamics. In this work, we combine those outputs with local observations using machine learning. So as to make the results usable for practitioners, we focus on simple and well known methods which can handle a high volume of data. We study first variable selection through two simple techniques, a linear one and a nonlinear one. Then we exploit those results to forecast wind speed and wind power still with an emphasis on linear models versus nonlinear ones. For the wind power prediction, we also compare the indirect approach (wind speed predictions passed through a power curve) and the indirect one (directly predict wind power).

Keywords—Wind speed forecasting, Wind energy forecasting, Machine learning, Numerical weather prediction, Downscaling

Abbreviations—Numerical weather prediction (NWP); machine learning (ML); European Centre for Medium-Range Weather Forecasts (ECMWF); parc de Bonneval (BO); parc de Moulin de Pierre (MP); parc de Beaumont (BM); parc de la Renardière (RE); parc de la Vènerie (VE); kernel ridge regression (KRR); neural network (NN); ordinary least squares (OLS); forward stepwise ordinary least square (OLS f-stepwise); reproducing kernel Hilbert space (RKHS); Hilbert-Schmidt independence criterion (HSIC); Backward selection with HSIC (BAHSIC); normalized root mean squared error (NRMSE)

1 Introduction

The fast development of renewable energies is a necessity to mitigate climate changes [Masson-Delmotte et al., 2021]. Wind energy has developed fast over the past three decades, with an average annual growth rate of 23.6% between 1990 and 2016 [IEA, 2018], and is now considered as a mature technology. It even maintained a significant growth in 2020 while global energy demand fell by 4% [IEA, 2021b]. The share of renewable energies in global electricity generation reached 29% in 2020, and is expected to keep growing fast in coming years [IEA, 2021a] which raises a number of challenges, stemming from the variability and spatial distribution of the resource. Then, in order to facilitate the dynamic management of electricity networks, forecasts of wind energy require continual improvement. Short timescales, from a few minutes to a few hours, are of particular importance for operations.

To produce forecasts, one can rely on several distinct sources of information. On timescales of half a day to about a week, deterministic weather forecasts provide a representation on a grid of the atmospheric state, including wind speed near the surface. The skill of such numerical weather forecasts (NWP) models has continually increased over the past decades [Bauer et al., 2015], while their spatial resolution has also grown finer (down to few km). However, on timescales shorter than a day, the use of such models is impeded by several difficulties, the two main ones being **(i)** the errors in the modeled wind and **(ii)** the relatively infrequent initiation of forecasts. The former result from limited resolution and poorly modeled processes. For instance a surface variable such as wind speed at an altitude of 100 m is strongly affected by local small-scale features and turbulent motions, both of which remain beyond the spatial resolution that is achievable for NWP models. Regarding the second point, operational centers typically launch forecasts twice or four times per day, however the calculation of the forecast itself as well as the preparation of its initial state

[a] LTCI, Télécom Paris, Institut Polytechnique de Paris

[b] LMD-IPSL, Ecole polytechnique - IP Paris, ENS - PSL Université, Sorbonne Université, CNRS, France

[c] Université de Lorraine, CNRS, IECL

[d] CMAP, Ecole Polytechnique, Institut Polytechnique de Paris

[*] Corresponding author: dimi.bouche@gmail.com; Office 5C06, Télécom Paris, 19 Place Marguerite Perey, 91120 Palaiseau.

require time and computational resources—see, e.g. [Kalnay, 2003].

As a result, methodologies for forecasting short-term wind speed or wind power at a site make use of past local observations and focus on statistical methods [Tascikaraoglu and Uzunoglu, 2014]. Indeed, a great variety of models and strategies have been explored for prediction horizons ranging from minutes to a few hours, and they leave aside potential inputs from NWP (e.g. Okumus and Dinler [2016], Liu et al. [2019]). Nevertheless, we know that NWP models can provide valuable information for the evolution of the atmosphere on larger scales—i.e. the formation and passage of a low-pressure system, and associated fronts. This raises several questions. Can local observations and NWP models be combined efficiently to improve short term forecasts ? How does the relative importance of those two sources of information evolve through time ?

To address those questions, we use both sources of information to train machine learning (ML) models. We then exploit the fact that local deficiencies in NWP models can partly be overcome by *downscaling*, i.e. better estimating local variables from the knowledge from a model's outputs and past observations. Such efforts have been carried out for decades in meteorology and climatology, under different names. In a pioneering early study, Glahn and Lowry [1972] applied multilinear regressions to correct NWP errors, after a training on past observations. More recently, such *model output statistics* has become common practice in operational weather forecast centers (e. g. Wilson and Vallée [2002], Baars and Mass [2005]). In recent years, ML methods have enhanced the performance of these post-processing steps (e.g. Zamo et al. [2016], Goutham et al. [2021]).

Dupré et al. [2020] showed for a wind farm in mainland France that: **(i)** the accuracy of the NWP model by itself begins to be competitive relative to that of statistical methods (using past local observations only) for times much shorter than a day (between one and two hours). **(ii)** A hybrid method combining a NWP model's output and past local measurements using ML can lead to an optimal forecast, which outperforms all methods and seamlessly makes the transition from one source of information to the other. The current study builds on and extends this work in several ways.

Many of the studies mentioned in Okumus and Dinler [2016], Liu et al. [2019] have some shortcomings. From a statistical perspective, the evaluation is generally performed only for one location and/or on data sets with relatively low number of samples while models are not evaluated using several splits of the dataset. The results may consequently be subject to variability. Then from a practical standpoint it may be hard to choose a method in practice because of the complex implementations for some, of the lack of comparisons for others, or because the propositions may not handle large datasets. Moreover, the behavior of the methods for several prediction horizons is generally not studied. Finally, many contributions focus either on wind power or wind speed prediction but not on the relation between them.

In the paper we propose to combine a NWP model's forecasts with past local observations to predict both wind speed and wind power. For the latter variable, we compare the direct and indirect (wind speed predictions passed through a power curve) approaches to prediction. As a first step, we perform variable selection and study the evolution of the importance of the selected variables through time. This allows us to better understand the nature of the studied statistical relationship and to extract a usable set of important variables. This also leads to an increase in the efficiency and scalability of the methods used downstream. A key focus is also to make this study usable for practitioners. To that end we concentrate on a reduced choice of efficient and well known ML methods which scale well with the number of samples or we use approximations that do, and moreover provide all elements for a straightforward implementation. All the comparisons are performed for several locations which enables us to expose many similarities but also some site specificities. Finally, all of the above elements' statistical significance is increased thanks to a thorough evaluation process: we study several sites over several periods of time (the number of samples is quite high per site) and for each location, we average the results over several splits of the dataset. To the best of our knowledge, no study combining all those elements has been published to date.

In Section 2, we introduce the data set from Zéphyr ENR and detail our processing of the data. Section 3 is dedicated to the presentation of our methodology as well as to the introduction of the statistical learning tools. Then in Section 4 we study practically two variables selection techniques—one linear and one nonlinear to draw insights on the important variables for wind speed and wind power prediction. Finally in Section 5 exploiting the variable selection techniques, we compare different well-known ML models and compare as well the direct and indirect strategies for the prediction of wind power.

Notation We introduce the following notation: for two integers $n_0, n_1 \in \mathbb{N}^*$, the set of strictly positive integers, we denote by $[\![n_0]\!]$ the set $\{1, ..., n_0\}$ and by $[\![n_0, n_1]\!]$ the set $\{n_0, ..., n_1\}$.

2 Data and context

In this Section, we introduce the dataset that we use (Section 2.1) as well as the pre-processing steps that we apply to it and the general evaluation methodology (Section 2.2).

2.1 Zéphyr ENR's dataset

The first source of information we use corresponds to measurements made by sensors in the wind turbines (we call those in situ variables) whereas the second one consists of forecasts from the European Centre for Medium-Range Weather Forecasts (ECMWF). We study five wind farms in the northern half of France:

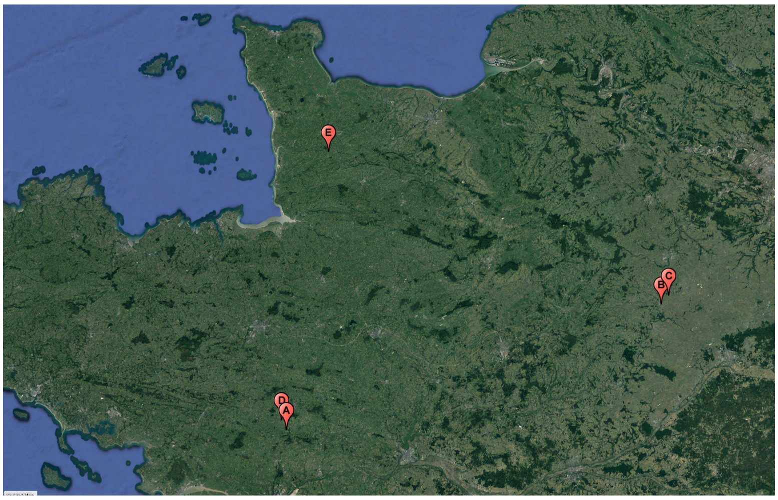


Figure 1: Cartography of the studied farms, BM (A), BO (B), MP (C), RE (D), VE (E)

Parc de Bonneval one (BO), Moulin de Pierre (MP), Parc de Beaumont (BM), Parc de la Renardière (RE), and Parc de la Vènerie (VE). Those wind farms are operated by the private company Zéphyr ENR and are described in details in [Dupré et al., 2020]. We display their location on a map in Figure 1. Some of those farms are geographically close by—we can form the pairs (BO, MP) and (BM, RE)—while VE is isolated. Note that another farm is available in the data, but we did not include it in the study because it displayed signs of sensors deficiencies. On the one hand, the geographical topology of the surroundings for (BO, MP) and (BM, RE) are quite similar, they correspond to open fields with very few elevation variations. On the other hand, VE is surrounded by wooded countryside with slightly more elevation variations, which may explain the differences that we observe between this farms and the others in Sections 4 and 5.

For BO and VE we have three years of data (from 2015 to 2017) which amounts to a total of $n = 157680$ observations for BO. However, for VE we do not use the year 2016 because it encompasses sensor deficiencies, so we use $n = 105120$ observations. For BM and RE we have access to two years of data (from 2017 to 2018 for BM and from 2015 to 2016 for RE) which results in a total of $n = 105120$ observations, and finally for MP we have only one year (2017), which gives us of total of $n = 52560$ observations. Several variables are available, we summarize the in situ ones in Table 1d—note that some variables are available only for certain farms. Also, in order to encode the circular nature of the in situ wind direction, when it is available, we encode it using two trigonometric variables.

The ECMWF provides several possible types of data, among which global forecasts issued by their NWP models. We followed [Dupré et al., 2020]: we extracted the day ahead forecast twice a day (at 0000UTC and 1200UTC) and included the same 47 variables as they do. Those variables are either selected or computed so as to describe as well as possible the boundary layer, the wind parameters and the temperature in the lower troposphere. For the sake of completeness, we display the details of those variables in Table 1. Table 1a presents the ECMWF surface variables, Table 1b the ECMWF altitude ones, while Table 1c displays the variables that are computed either from surface or altitude variables. The spatial resolution of ECMWF forecasts is about 16 km (0.125 ° in latitude and longitude) and their temporal resolution is 1h, then to match that of the in situ variables (10 min), we linearly interpolate the ECMWF forecasts. To finish with we sum up the abbreviations for the variables mostly used in the paper in Table 1e.

2.2 Preprocessing and evaluation methodology

In order to increase the statistical significance of the results, for each wind farm, we average the outcomes of the experiments over different splits of the dataset. Concretely, a split consists of 3 subsets from the dataset, a train subset (of size n_{train}), a validation one (size n_{val}) and a test one (of size n_{test}). We train the models on the train set, select their parameters on the validation set and evaluate their performance on the the test set. We build such subsets in a rolling fashion so as to preserve time coherence. For instance for the first split we take the period $[\![n_{\text{train}}]\!]$ for training, the period $[\![n_{\text{train}} + 1, n_{\text{train}} + n_{\text{val}}]\!]$ for validation and we test the models on the period $[\![n_{\text{train}} + n_{\text{val}} + 1, n_{\text{train}} + n_{\text{val}} + n_{\text{test}}]\!]$. Then for the second split, the train period is $[\![n_{\text{train}}+n_{\text{val}}+n_{\text{test}}+1, 2n_{\text{train}}+n_{\text{val}}+n_{\text{test}}]\!]$, the validation one is $[\![2n_{\text{train}}+n_{\text{val}}+n_{\text{test}}+1, 2n_{\text{train}}+2n_{\text{val}}+n_{\text{test}}]\!]$ and so on... For the sizes of the windows, we set $n_{\text{train}} = 10000$, $n_{\text{val}} = 10000$ and $n_{\text{test}} = 10000$ (however, the last split generally contains around $5000 \leq n_{\text{test}} \leq 10000$ observations). Since the length of available data vary from farm to farm, we do not have the same number of splits for all the farms.

We pre-process the data in the following way. As the number of wind turbines per farm is quite low (6 for BM, 6 for BO, 3 for HC, 6 for MP, 6 for RE and 4 for VE) and so as to smooth out some segments which encompass missing data, we average the in situ data over the turbines for each farm. In all our experiments, we standardize both the input and the output variables (subtract the mean and divide by the standard deviation), using the available training data (the training data from the currently used split). We do so

Altitude	Variable	Unit
10m/100m	Zonal wind speed	ms^{-1}
	Meridional wind speed	ms^{-1}
2m	Temperature	K
	Dew point temperature	K
Surface	Skin temperature	K
	Mean sea level pressure	Pa
	Surface pressure	Pa
	Surface latent heat flux	Jm^{-2}
	Surface sensible heat flux	Jm^{-2}
	Boundary layer dissipation	Jm^{-2}
	Boundary layer height	m

(a) ECMWF surface variables

Pressure level (hPa)	Variable	Unit
1000/925/850/700/500	Zonal wind speed	ms^{-1}
	Meridional wind speed	ms^{-1}
	Geopotential height	m^{2}s^{-2}
	Divergence	s^{-1}
	Vorticity	s^{-1}
	Temperature	K

(b) ECMWF altitude variables

Altitude	Variable	Unit
10m/100m	Norm of wind speed	ms^{-1}
10m to 925 hPa	Wind shear	ms^{-1}
	Temperature gradient	K

(c) ECMWF computed variables

Availability	Variable	Unit
All	Wind speed	ms $^{-1}$
All	Power output	kW
All but HC	Wind direction	Degree
BO and BM	Temperature	Celsius degree

(d) In situ variables

Variable (source)	Abbreviation
Wind speed (in situ)	WS
Power output (in situ)	PW
Norm of wind speed at 100m (ECMWF)	F10
Norm of wind speed at 100m (ECMWF)	F100
Wind shear between 10m and 925 hPa (ECMWF)	DF
Boundary layer dissipation (ECMWF)	bld
Boundary layer height (ECMWF)	blh
Surface latent heat flux (ECMWF)	slhf

(e) Abbreviations for the variables used in the paper

Table 1: Variables used in our analysis

both for in situ variables and ECMWF ones. Such operation is crucial for instance to avoid favoring some variables which are structurally bigger over other when using regularized machine learning models.

3 Methodology and machine learning tools

In this Section, we introduce our general methodology as well as the ML tools that we use for variable selection and forecasting for wind speed and wind power prediction.

3.1 Methodology

Dataset building Let $m \in \mathbb{N}$ be the prediction length (the number of future wind speed or wind power values we want to forecast). For the ECMWF variables, we include the corresponding forecasts. However, in practice we found that including a bit more than that improved forecasting quality. To that end, we denote respectively by $r_0 \in \mathbb{N}$ the number of past ECMWF predictions that we include, and by $r_1 \in \mathbb{N}$ the number of ECMWF predictions that we consider after m. For the in situ variables, we include a length $l \in \mathbb{N}$ of past observations. Those different time windows are illustrated in Figure 2 (considering a reduced number of variables). For the different variables and time windows, we concatenate the relevant observations $\mathbf{x}_t$ (those within the orange zones in Figure 2). From those, we want to produce a prediction for $\mathbf{y}_{t+1:m} \in \mathbb{R}^m$ (the m observations within the green zone in Figure 2). In practice, we use the following parameters which work well experimentally:

- we predict up to 4 hours, with a time sampling rate of 10 min, it means that $m = \frac{4\times60}{10} = 24$,
- for the in situ variables, we consider 3h of past observations, thus $l = \frac{3\times60}{10} = 18$,
- for the ECMWF variables we additionnally use the predictions between 1.5h before and 1.5h after the time horizon of interest so $r_0 = r_1 = \frac{1.5\times60}{10}$.

ML models We compare several ML models. We stick to ones which are well-known and can scale well to a higher volume of data. Good results were obtained for one location (BO) from the studied dataset in Dupré et al. [2020] using linear regression with greedy forward stepwise variable selection [Hastie et al., 2001], so this method consists of our first benchmark. Nevertheless, since we are particularly interested in the importance

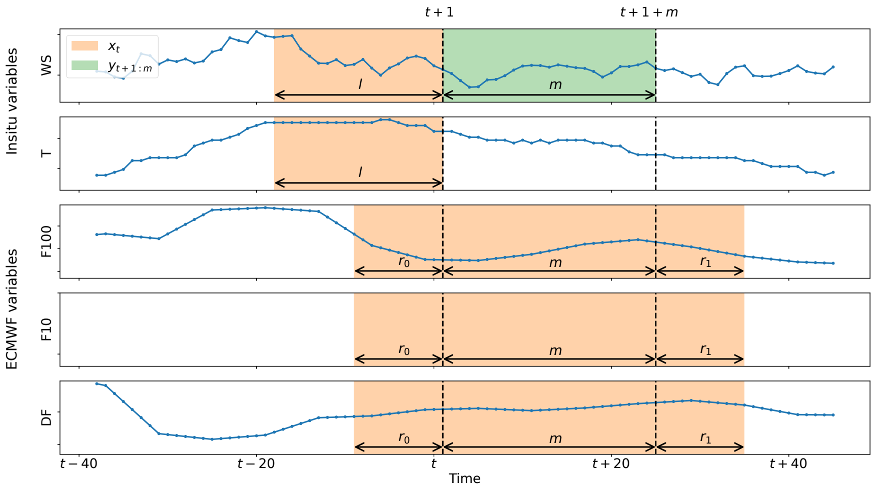


Figure 2: Summary of the time windows used for each source of data for wind speed prediction

of variables, we study also an alternative which select variables directly in the least square problem. The LASSO [Tibshirani, 1996] exploits the L1 penalty (sum of the absolute values of the coefficients) to induce sparsity in the regression coefficients, thus shrinking to zero those which are associated to the less relevant variables. Such methods can however be limited in that they can learn only linear dependencies between the input and output variables. Consequently, we study a principled nonlinear alternative: kernel ridge regression (KRR)—see for instance [Schölkopf and Smola, 2002, Shawe-Taylor and Cristianini, 2004]. Finally, in order to include most families of ML models, we consider two other nonlinear methods: a tree-based boosting algorithm [Friedman, 2001] (we use XG-Boost Chen and Guestrin [2016]) as well as a feed-forward neural network (NN). In Section 3.2 we give more mathematical details on the general ML problem as well as on the methods that perform the best in the experimental section.

Variable selection We have many in situ and ECMWF variables available (see Table 1). So as to improve the computational efficiency and and understand better what the models do, it is preferable to use only the most important variables. Ideally, we want to find a subset which is sufficient for a model to capture most of the relevant statistical relationship between the input variables and the target. In that sense a variable selection tool is necessarily model specific. Linear techniques will focus only on linear dependencies, whereas nonlinear ones will incorporate a much wider range of dependencies. The former are thus adapted to linear models whereas the latter are adapted to a wider class of models. Then we propose to use and interpret the results of one variable selection for each type. For the linear one we study the LASSO. For the nonlinear one, we use backward elimination using the Hilbert Schmidt Independence Criterion [Song et al., 2012]. As opposed to the LASSO, it performs variable selection as an independent first step. The selected variables can then be used downstream to train any model. Then, for the nonlinear models (KRR, XG-Boost, feed-forward NN), we use the variables selected through this method. We give more detailed insights into the different techniques in Section 3.3.

3.2 Details on machine learning models

The input observations are the concatenation of the different variables on the time windows described in the previous section. We denote by $\mathcal{X} = \mathbb{R}^q$ the resulting input space, for some $q \in \mathbb{N}$ depending on the chosen variables and time windows. Our training data then consist of $((\mathbf{x}_t, \mathbf{y}_{t+1:m}))_{t=1}^n \in (\mathcal{X} \times \mathbb{R}^m)^n$ for some $n \in \mathbb{N}$, where we recall that $\mathbf{y}_{t+1:m} = (y_{t+1+m_0})_{m_0=1}^m$.

Given a prediction function from a ML model class $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ parameterized by a vector $\mathbf{w} \in \mathcal{W}$, we want to minimize the average error on the training data:

$$\min_{\mathbf{w}\in\mathcal{W}} \frac{1}{n} \sum_{t=1}^{n} \|f_{\mathbf{w}}(\mathbf{x}_t) - \mathbf{y}_{t+1:m}\|_2^2 + \lambda\Omega(\mathbf{w}). \tag{1}$$

However, depending on the model, a penalty function $\Omega : \mathcal{W} \longrightarrow \mathbb{R}$ can be added in order to prevent overfitting of the training data or promote variable selection; its intensity is controlled by a parameter $\lambda > 0$.

In practice, instead of predicting all time horizons in one go as in Problem (1), we rather use separate models for each horizon in $[\![t+1, t+1+m]\!]$. That way we can tailor the different parameters for each horizon, which we found to improve performances. Then in what follows, we consider a generic time horizon m and predict y_{t+1+m}.

Ordinary least squares (OLS) In forward stepwise variable selection [Hastie et al., 2001], at each step an OLS regression is solved for which the optimization problem reads:

$$\min_{\mathbf{w}\in\mathcal{W},b\in\mathbb{R}} \frac{1}{n}\sum_{t=1}^{n}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_t + b - y_{t+1+m})^2 \tag{2}$$

A well-known and simple closed form exist, which we use in practice.

LASSO. The optimization problem for the LASSO is the following:

$$\min_{\mathbf{w}\in\mathcal{W},b\in\mathbb{R}} \frac{1}{n}\sum_{t=1}^{n}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_t + b - y_{t+1+m})^2 + \lambda\|\mathbf{w}\|_1,$$

where $\|\mathbf{w}\|_1$ is the sum of the absolute values of the coefficients $\mathbf{w}$. Many efficient algorithms exist to solve this convex yet non differentiable problem—see for instance [Beck and Teboulle, 2009]. In practice we use the scikit-learn [Pedregosa et al., 2011] implementation with coordinate descent solver.

Kernel ridge regression (KRR). In KRR, we consider a class of models defined by a positive definite reproducing kernel $k : \mathcal{X}\times\mathcal{X}\longrightarrow\mathbb{R}$ which results in a unique associated reproducing kernel Hilbert space (RKHS). A most typical choice for k is the Gaussian kernel:

$$k_\gamma(\mathbf{x},\mathbf{x}') := \exp\left(-\gamma(\|\mathbf{x}-\mathbf{x}'\|_2^2\right).$$

We then seek our modeling function in this RKHS which we denote $\mathcal{H}_k$, each $h\in\mathcal{H}_k$ being a function from $\mathcal{X}$ to $\mathbb{R}$. For many kernels, this space constitutes a very rich class of modeling functions which can model nonlinear dependencies. The optimization problem reads:

$$\min_{h\in\mathcal{H}_k} \frac{1}{n}\sum_{t=1}^{n}(h(\mathbf{x}_t) - y_{t+1+m})^2 + \lambda\|h\|_{\mathcal{H}_k}^2 \tag{3}$$

where $\|\cdot\|_{\mathcal{H}_k}^2$ is the RKHS norm on $\mathcal{H}_k$, which measure in a sense the smoothness of functions in $\mathcal{H}_k$. Thanks to the Representer theorem, any solution to Problem (3) can be parameterized by a vector $\boldsymbol{\alpha}\in\mathbb{R}^n$:

$$h_{\boldsymbol{\alpha}} := \sum_{j=1}^{n}\alpha_j k(\mathbf{x}_j,\cdot),$$

which makes optimization in the RKHS amenable. For KRR the optimal coefficients $\widehat{\boldsymbol{\alpha}}$ can be found in close form:

$$\widehat{\boldsymbol{\alpha}} := (K + n\lambda I)^{-1}\mathbf{y}^{(m)},$$

with $\mathbf{y}^{(m)} := (y_{t+1+m})_{t=1}^n$, $I\in\mathbb{R}^{n\times n}$ the identity matrix and $K\in\mathbb{R}^{n\times n}$ with entries $K_{tj} = k(\mathbf{x}_t,\mathbf{x}_j)$.

In practice, to handle the large volume of training data, we use an approximated version of KRR. Nyström approximation [Williams and Seeger, 2001, Drineas and Mahoney W., 2005] exploits a random subset of points from the training data. Concretely, we sample randomly and uniformly without replacement $p\in\mathbb{N}$ indices $\{i_1, ..., i_p\}$ among the integers in $[\![n]\!]$, and replace $h_{\widehat{\boldsymbol{\alpha}}}$—see e.g. [Rudi et al., 2015]—with:

$$\widetilde{h}_{\widetilde{\boldsymbol{\alpha}}} := \sum_{j=1}^{p}\widetilde{\alpha}_j k(\mathbf{x}_{i_j},\cdot),$$

where $\widetilde{\boldsymbol{\alpha}}\in\mathbb{R}^p$ is given by the following close form:

$$\widetilde{\boldsymbol{\alpha}} := (K_{np}^{\mathrm{T}}K_{np} + \lambda n K_{pp})^{\dagger}K_{np}^{\mathrm{T}}\mathbf{y}^{(m)}. \tag{4}$$

Where $A^\dagger$ denotes the Moore-Penrose pseudo-inverse of a matrix A, and $K_{np}\in\mathbb{R}^{n\times p}$ is defined by the entries $(K_{np})_{tj} := k(\mathbf{x}_t,\mathbf{x}_{i_j})$ and $K_{pp}\in\mathbb{R}^{p\times p}$ by the entries $(K_{pp})_{bj} = k(\mathbf{x}_{i_b},\mathbf{x}_{i_j})$.

3.3 Details on variable selection

OLS with forward stepwise selection (OLS f-stepwise) When performing linear regression, variable selection can be performed in a greedy manner. First an intercept is fit to the data and then at each step we solve OLS problems—Problem (2)—adding in turns each one of the remaining variables. We then keep the one which best improve the model according to some criterion. In [Dupré et al., 2020], the Bayesian information criterion is used. However, in our experiments we rather used the improvement of the score on half of the validation set, as it led to better experimental performances.

LASSO Provided the regularization intensity λ is well chosen, the L1 penalty of the LASSO shrinks the coefficients associated the variables which are less important towards zero. Then the model uses mostly the relevant variables and the magnitude of the coefficients can be looked at to deduce what those variables are. This is the type of analysis that we perform in Section 4.1.

Hilbert-Schmidt independence criterion (HSIC). The HSIC [Gretton et al., 2005] is an independence measure. Similarly to the KRR, it makes use of RKHSs to embed implicitly a set of observations into a high-dimensional space and consider a notion of independence in this space which allows for detection of nonlinear dependencies. More precisely, let us consider a positive-definite kernel $k : \mathcal{X}^2 \longrightarrow \mathbb{R}$ for the input observations and a one $g : (\mathbb{R}^m)^2 \longrightarrow \mathbb{R}$ for the output observations. For this variable selection technique, we consider all time horizons in $[\![t+1, t+1+m]\!]$ together as the kernelized framework allows for this. In practice, we estimate HSIC from the data as [Gretton et al., 2008]:

$$\widehat{\text{HSIC}} := \frac{1}{n^2}\text{Trace}(HKHG),$$

where $H \in \mathbb{R}^{n\times n}$ is the centering matrix $H := \frac{1}{n}(I - \mathbf{1}\mathbf{1}^{\mathrm{T}})$ with $\mathbf{1} \in \mathbb{R}^n$ a vector full of ones and $I \in \mathbb{R}^{n\times n}$ the identity matrix. The matrices $K \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times n}$ are the kernel matrices:

$$\begin{aligned}(K)_{tj} &:= k(\mathbf{x}_t, \mathbf{x}_j),\\ (G)_{tj} &:= g(\mathbf{y}_{t+1:m}, \mathbf{y}_{j+1:m}).\end{aligned}$$

HSIC takes its values between 0 and 1, a value of 0 meaning independence and a value of 1 means full dependence.

However, to be able to compute the estimator for the large number of data points, we recourse to Nyström approximation as well [Zhang et al., 2018]. We then sample randomly and uniformly without replacement $p \in \mathbb{N}$ indices $\{i_1, ..., i_p\}$ from the integers in $[\![n]\!]$ for the input observations and $p' \in \mathbb{N}$ ones $\{i'_1, ..., i'_{p'}\}$ for the output observations. We then define the Nyström features maps (centered in the feature space using H):

$$\begin{aligned}\widehat{\Phi} &:= HK_{np}K_{pp}^{-\frac{1}{2}},\\ \widehat{\Psi} &:= HG_{np'}G_{p'p'}^{-\frac{1}{2}},\end{aligned}$$

where the matrices K_{np} and K_{pp} are defined as for Equation (4) and the matrices $G_{np'}$ and $G_{p'p'}$ are defined similarly for the kernel g however based on the set of indices $\{i'_1, ..., i'_{p'}\}$.

The Nyström HSIC estimator is then [Zhang et al., 2018]:

$$\widetilde{\widehat{\text{HSIC}}} := \|\frac{1}{n}\widehat{\Phi}^{\mathrm{T}}\widehat{\Psi}\|_F^2,$$

where for a matrix A, the Frobenius norm is defined as $\|A\|_F^2 := \text{Trace}(A^{\mathrm{T}}A)$.

Backward selection with HSIC (BAHSIC) To perform variable selection, we start with all the available variables and then at each round, we compute the HSICs between the input variables and the target variable removing one input variable at a time. A given percentage of the input variables for which those HSICs are the highest are removed. We keep iterating in this way to rank the variables. Then, the ones removed the latest are the most important ones. The detailed algorithm corresponds to Algorithm 1 in [Song et al., 2012]. A forward version exists as well, however, the authors advocate the use of backward selection to avoid missing important variables. Finally as a side note, in practice we use as Gaussian kernels setting bandwidth following the recommendations from [Song et al., 2012].

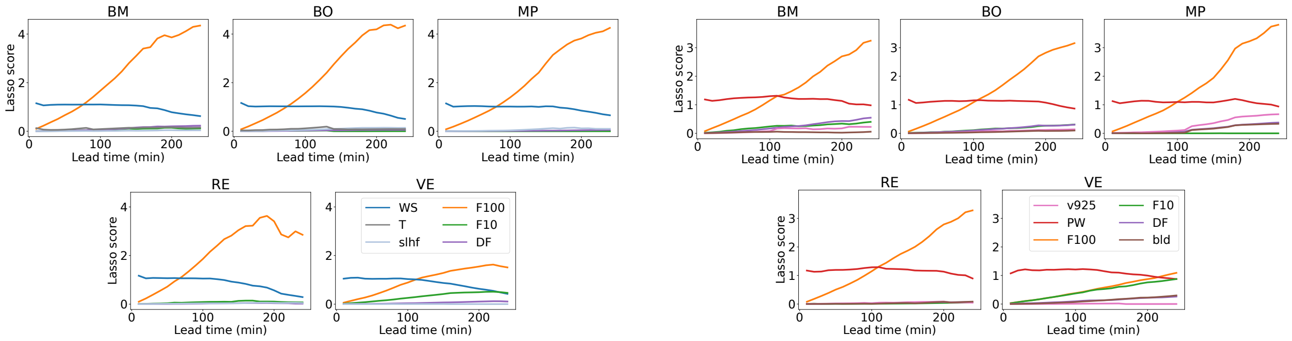
4 Importance of variables and their evolution through time

In this section, we study variable selection using the LASSO in Section 4.1 and BAHSIC in Section 4.2 to determine which variables are the most important and how their importance evolves through time.

4.1 Linear variable selection with LASSO

LASSO scores We now describe the computations carried out to extract a subset of relevant variables suitable for interpretation from fitted LASSO models. In practice for each data split, we validated the regularization parameter λ on the validation set and obtained estimated LASSO coefficients. Now to reduce the number of variables we must rank them according to an importance metric based on those coefficients; we do so for each time horizon separately. So as to to avoid assigning more weight to models for which λ was selected small, for each farm and each data split we normalize the coefficients by the absolute value of the biggest one. As opposed to the grouped variable selection performed in Section 4.2, the observations through time for a given variable can be separated by the LASSO shrinking (a variable can be selected for instance at time t_0 but not at t_1). Consequently, we sum the normalized coefficients corresponding to different time instants for the same variable and in doing so, we obtain a single quantity per variable. Then we average those quantities over the data splits and call the resulting quantities LASSO scores. To sum up, at this point we have for each prediction horizon and each farm a set of such scores for each variable. Then, to select variables, we average those LASSO scores over farms. Finally, based on those average scores, for each prediction horizon, we keep the top 6 variables.

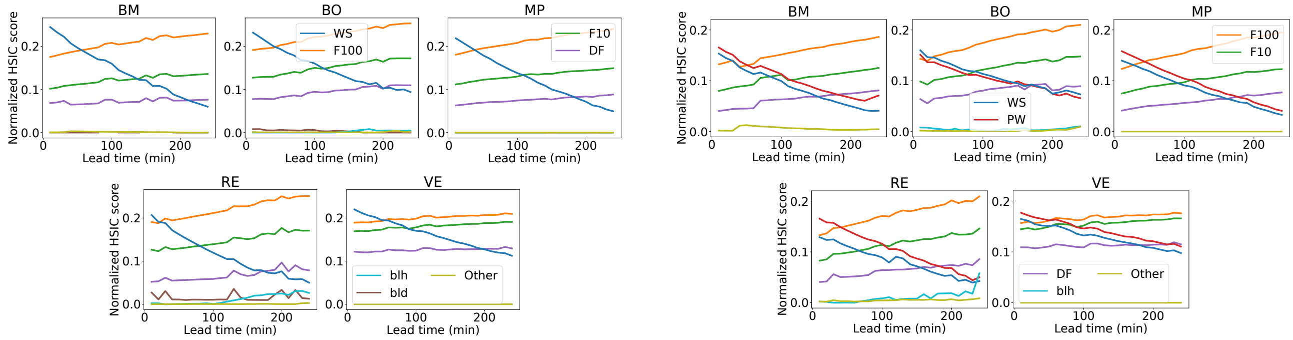
Interpretation For those selected variables, we plot the evolution through time of the LASSO scores in Figure 3. We consider both targets—wind speed in Figure 3a and wind power in Figure 3b. We make the following key observations:



(a) Wind speed as target variable

(b) Wind power as target variable

Figure 3: Linear variable selection with the LASSO



(a) Wind speed as target variable

(b) Wind power as target variable

Figure 4: Nonlinear variable selection using HSIC

- At all locations, two variables are much more important than the rest. The in situ observed wind speeds (WS) and the ECMWF predicted wind speed at altitude 100 m (F100) stand out for wind speed prediction, and for wind power prediction, in situ power production (PW) along with F100 are of particular importance. We can relate those results to the good performances of the LASSO for wind speed prediction in Section 5. Then if we look at the relative magnitudes of the coefficients, we can deduce that only using a linear combination of past local wind speeds (WS) and predicted wind speeds (F100), we can get an already good description of the future local wind speed.
- The location VE can be singled out from the others. Indeed the predicted wind speed at altitude 10 m (F10) appears, and the forecasted wind speeds (F100 and F10) take longer to take over the in situ variables, especially when predicting wind power. This may be explained by a lesser representativity of the ECMWF forecasts for this location which may be linked to the elevation variations in the surroundings of the farm that we mentioned in Section 2.1.

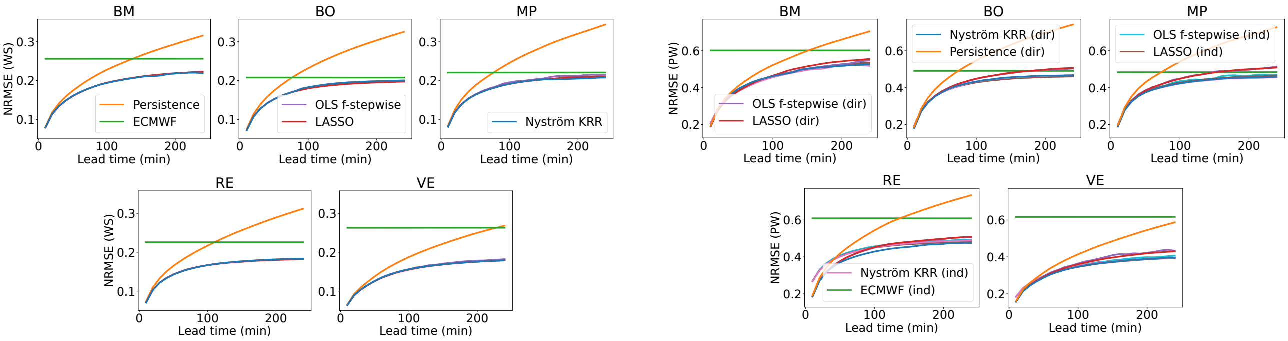
As a concluding note, the dynamics of the local wind speed seem to be very well approximated by a simple linear model combining very few in situ variables and ECMWF ones. For predicting directly wind power however, we see in Section 5 the results are a bit less convincing, possibly due to the nonlinear aspect of the power curve.

4.2 Nonlinear variable selection with HSIC

HSIC-score For each farm and each split of the data we run BASHSIC on the training set until we have 5 variables left. Then for each variable we estimate the HSIC with that variable removed and normalize this value using the maximum HSIC value among those quantities. The normalized HSIC score appearing in the figures corresponds to 1 minus this score averaged over the training sets; the higher it is, the more important the variable is. We display the results with the two possible different targets in Figure 4 (wind speed in Figure 4a and wind power in Figure 4b).

Global interpretation We make the following key observations:

- As expected, the in situ variables are most relevant for the shortest time horizon and the ECMWF variables take progressively the lead for longer horizons. However, compared to linear feature selection ECMWF variables take the lead faster here—between 10-50 minutes as opposed to 70-100 minutes for linear selection. The retained variables are mostly the same as the one selected by the LASSO (WS or WS and PW depending on the target) and F100. However, F10 and DF are now more systemically retained with a significant importance.



(a) Wind speed as target variable

(b) Wind power as target variable

Figure 5: Overview of NRSMEs for the best performing methods

- As for the LASSO selection, we can single out VE where the importance of the in situ variable(s) decreases much less fast than at the other farms.

Presence of DF and F10. In comparison with linear selection, we have two more variables of interest (DF and F10). F10 describes the wind speed at lower levels and DF the wind shear near the surface. They thus bring useful information about the wind and its vertical shear, and likely help to correct deficiencies of the NWP model's description of wind at 100m. The fact that they appear here and not in the linear framework indicates a nonlinear relation, which is not surprising as the shear relates to the level of turbulence in the boundary layer. Additionally, the above results bring a fairly sharp answer to another question underlying this study. As the calculation of near-surface winds in NWP models involves parameterizations, they are not the most reliable output of NWP models. Consequently, one could expect that, informed about other aspects of the boundary layer and local wind realizations, a nonlinear method could capture better the relationship between the boundary layer and the near-surface winds. This is not the case: BAHSIC clearly select rather the wind variables as the best source of information. Over variable terrain (VE), wind speed at different heights (F10) are more used, suggesting that the NWP model indeed fails to accurately describe the wind shear. And yet variables describing the boundary layer (e.g. stratification) still remain unused or marginal. Over flat terrain, the wind speed at 100m height (F100) is the major source of information, which is positive and encouraging regarding the accuracy of NWP models.

5 Wind speed and wind power forecasting

In this Section we compare several ML models for predicting both wind speed and wind power, exploiting the variable selection techniques from the previous section. We include the two main baselines, namely persistence which predicts the last in situ observation and ECMWF which uses the F100 forecasts from the ECMWF.

5.1 Experimental setup

We evaluate our results using the normalized root mean squared error (NRMSE) as in [Dupré et al., 2020]. Let $(z_t)_{t=1}^n$ denote the realizations of a (scalar-valued) target variable. We define its global mean as $\bar{z} = \frac{1}{n}\sum_{t=1}^n z_t$. Given a set of predicted values $(\widehat{z}_t)_{t=1}^n$, it is defined as:

$$\text{NRMSE} := \frac{\sqrt{\frac{1}{n}\sum_{t=1}^n(\widehat{z}_t - z_t)^2}}{\bar{z}}.$$

We follow the methodology introduced in Section 3.1 and refer the reader to Section 3.2 for details on the ML methods. In practice, for each data split, the key parameters of the different methods are chosen using the validation set (the regularization parameters λ, the Gaussian kernel's γ for KRR, the number of variables for OLS f-stepwise, the architecture for the feedforward NN...). We provide the details of the considered parameters in the supplementary material and will make the code available.

Direct/indirect wind power prediction When we predict wind power, we consider two prediction techniques. Either we predict directly the wind power (direct approach) or we predict the wind speed which we transform using an estimated power curve in the same fashion as in [Dupré et al., 2020] (indirect approach). A theoretical power curve could be used as well, however, in this work we estimate it from the training WS and PW observations using median of nearest neighbors interpolation using 250 neighbors.

5.2 Comparison of forecasting methods

From a general perspective, our experiments show that combining a NWP model's outputs with local observations is very beneficial for predicting both wind speed and wind power at all the time horizons considered.

Method (average rank)	BM	BO	MP	RE	VE
LASSO (1.6)	1.14	**0.13**	0.17	**0.14**	**0.15**
Nyström KRR (2.0)	**1.07**	0.36	**0.07**	0.19	0.21
OLS f-stepwise (2.4)	1.12	0.16	0.48	0.17	0.34
Feedforward NN (4.4)	1.54	0.87	0.61	1.18	1.34
XG-Boost (4.6)	1.83	1.25	1.64	0.73	0.97
ECMWF (6.4)	7.83	3.63	3.76	6.65	11.37
Persistence (6.6)	5.61	6.72	7.03	6.7	4.5

Table 2: Average NRMSE degradation w.r.t. best predictor for wind speed prediction ($\times 10^{-2}$)

Prediction type	Method (average rank)	BM	BO	MP	RE	VE
Direct	**Nyström KRR (2.2)**	3.93	1.16	0.63	**0.36**	**0.69**
Indirect	Nyström KRR (2.8)	**3.46**	1.15	**0.55**	2.88	1.2
Indirect	LASSO (3.0)	3.96	**0.77**	1.08	2.62	0.86
Indirect	OLS f-stepwise (4.0)	3.62	0.83	1.53	3.32	1.61
Direct	XG-Boost (4.4)	4.54	2.36	1.5	1.7	1.8
Direct	OLS f-stepwise (5.6)	4.07	2.98	3.56	2.33	3.0
Direct	LASSO (6.0)	4.83	3.12	3.46	2.44	2.45
Direct	Persistence (8.8)	12.0	15.45	14.91	14.16	9.88
Direct	Feedforward NN (9.6)	25.06	7.8	5.09	20.62	18.67
Indirect	Persistence (9.8)	13.13	15.86	15.22	16.29	10.85
Indirect	ECMWF (9.8)	18.85	8.63	8.02	19.64	28.53

Table 3: Average NRMSE degradation w.r.t. best predictor for wind power prediction ($\times 10^{-2}$)

To that end, Figure 5 displays the evolution of the NRSME for the two baselines as well as for the ML methods which performed best overall (Nyström KRR, LASSO, OLS f-stepwise). For wind power prediction, we add the results for direct and indirect predictions. In more details, for three farms (VE and to a lesser extend, BM and RE), even after four hours the improvement over ECMWF is still quite large. For BO and MP it becomes less important, yet still present. The improvement can be quite dramatic for very short horizons (first 100 minutes or so), and a bit less important for longer time horizons. This is probably linked to the representativity of the NWP model's outputs which depends on the location.

In order to compare quantitatively the ML methods, we introduce a new metric. Let $\mathcal{F}$ be a given set of predictors—for instance when predicting wind speed we have $\mathcal{F} := \{$Nyström KRR, LASSO, OLS f-stepwise, XG-Boost, Feedforward NN, Persistence, ECMWF$\}$. Given a predictor $f \in \mathcal{F}$, a prediction horizon $m_0 \in [\![m]\!]$ and a data split s (among a total of $S \in \mathbb{N}$ data splits), let $\mathrm{NRMSE}_{s,m_0}^{(f)}$ denote the NRMSE of predictor f for the prediction horizon m_0 on the data split s. We then define the average NRMSE degradation of a predictor f_0 (with respect to the best predictor):

$$\frac{1}{mS}\sum_{s=1}^{S}\sum_{m_0=1}^{m}\left(\mathrm{NRMSE}_{s,m_0}^{(f_0)} - \min_{f\in\mathcal{F}}\mathrm{NRMSE}_{s,m_0}^{(f)}\right).$$

The best possible value is zero as it means that over all splits and over all horizons, the predictor was the best one. We compare methods with this metric in Table 2 (WS as target) and in Table 3 (PW as target).

On the one hand, for wind speed prediction, the LASSO is the best ranked method. Relating this to the results from Section 4.1, it shows that the dynamics of the wind speed can be very well described by a linear combination of few ECMWF and local variables (essentially past local wind speeds and forecasted wind speeds). It suggests that the important nonlinear dynamics are overall well captured in the ECMWF variables, and as a consequence a simple linear model is sufficient. On the other hand, it seems better to predict directly wind power and do so using the Nyström KRR. This is not surprising, as the power curve is a nonlinear function and so we expected the linear methods to struggle in direct prediction. Moreover, in direct prediction, we implicitly include the power curve into the learning problem. Consequently, the importance of the different wind speeds on the power curve are taken into account. For instance, a model trained to predict wind speed first will be very eager to forecast well high values (failing to do so would incur a high error term). However to predict wind power, producing accurate forecasts for higher wind speeds is less critical, since in the power curve, the actual wind power as a function of the wind speed is thresholded.

Nevertheless, not all the nonlinear method performs well; XG-Boost and the feedforward NN do not. For the latter it may be explained by the optimization variability/error to which NN are subject. This is coherent, since the indirect LASSO is still a very good baseline, the much higher expressiveness of NNs may not be that big of an advantage here. Moreover, the direct approach with Nyström KRR is the best performing method. That latter method is nonlinear yet the learning problem is strongly convex thanks to regularization and a closed form solution exists, consequently its optimization error is close to non existent. This may constitute a big advantage on this problem where expressiveness beyond the handling of the nonlinearity of the power curve does not seem necessary.

6 Conclusion

We showed through experiments on several wind farms that we can improve very significantly short-term local forecasts of both wind speed and wind power by combining statistically a NWP model's outputs with local observations. To better understand how, we studied in details the evolution of the variables' importance using two metrics, a linear one based on LASSO coefficients and a nonlinear one using HSIC. Our global conclusion is that NWP's wind variables are a very relevant source of information to complement local observations, even for the very short-term. More precisely, to forecast wind speed, a parsimonious linear combination of NWP and local variables (with the LASSO) yielded the best result, while for wind power forecasting, the use of a nonlinear method (Nyström KRR) considering few variables (selected with BAHSIC) to predict directly the variable of interest (no power curve involved) is preferable. However, the main advantage of this more expressive model seems to come mostly from the ability to capture the nonlinearity of the power curve, rather than from its grasping of more complex dynamics, which the NWP model's outputs seem to describe sufficiently. For future work, assessing the variability of the predictions should be investigated. Instead of predicting conditional means as we do in this work, we could predict conditional quantiles [Koenker and Hallock, 2001] which would inform us on the expected distribution of the predictions. This could help mitigate the intermittent effects of wind power production further.

Acknowledgments

This work received support from the Télécom Paris research chair on Data Science and Artificial Intelligence for Digitalized Industry and Services (DSAIDIS). It was conducted as part of the Energy4Climate Interdisciplinary Center (E4C) of Institut Polytechnique de Paris and Ecole des Ponts ParisTech. It has received support as well from the 3rd Programme d'Investissements d'Avenir [ANR-18-EUR-0006-02] and from the Foundation of Ecole polytechnique (Chaire "Défis Technologiques pour une Énergie Responsable"). The authors would like to acknowledge the company Zéphyr ENR, and especially its CEO Christian Briard, for providing the data from its wind farms.

References

J.A. Baars and C.F. Mass. Performance of National Weather Service forecasts compared to operational, consensus and weighted model output statistics. *Wea. Forecast.*, 20:1034–1047, 2005. doi: 10.1175/WAF896.1.

P. Bauer, A. Thorpe, and G. Brunet. The quiet revolution of numerical weather prediction. *Nature*, 525: 47–55, 2015. doi: 10.1038/nature14956.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

Petros Drineas and Michael Mahoney W. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

Aurore Dupré, Philippe Drobinski, Bastien Alonzo, Jordi Badosa, Christian Briard, and Riwal Plougonven. Sub-hourly forecasting of wind speed and wind energy. *Renewable Energy*, 145:2373–2379, 2020.

Aurore Dupré, Philippe Drobinski, Jordi Badosa, Christian Briard, and Peter Tankov. The economic value of wind energy nowcasting. *Energies*, 13(20):5266, 2020.

Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

H.R. Glahn and D.A. Lowry. The use of model output statistics (MOS) in objective weather forecasting. *J. App. Meteor.*, 11:1203–1211, 1972.

N. Goutham, B. Alonzo, A. Dupré, R. Plougonven, R. Doctors, L. Liao, M. Mougeot, A; Fischer, and P. Drobinski. Using machine-learning methods to improve surface wind speed from the outputs of a numerical weather prediction model. *Boundary Layer Meteorology*, 2021. doi: 10.1007/s10546-020-00586-x.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *Algorithmic Learning Theory*, pages 63–77. Springer Berlin Heidelberg, 2005.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. *The Elements of Statistical Learning.* Springer, 2001.

IEA. Renewables information. *International Energy Agency*, page 12, 2018.

IEA. Renewables: Analysis and forecast to 2026. *International Energy Agency*, page 175, 2021a.

IEA. Global energy review 2021. *International Energy Agency*, page 36, 2021b.

Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability.* Cambridge University Press, 2003.

Roger Koenker and Kevin F. Hallock. Quantile regression. *Journal of Economic Perspectives*, 15(4):143–156, December 2001.

Hui Liu, Chao Chen, Xinwei Lv, Xing Wu, and Min Liu. Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods. *Energy Conversion and Management*, 195:328–345, September 2019. doi: 10.1016/j.enconman.2019.05.020.

V. Masson-Delmotte, P. Zhai, A. Pirani, S.L. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. I. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T. K. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. Zhou (eds.). IPCC, 2021: Sumarry for Policymakers. In *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*, pages 1–42, 2021.

Inci Okumus and Ali Dinler. Current status of wind energy forecasting and a hybrid method for hourly predictions. *Energy Conversion and Management*, 123:362–371, September 2016. doi: 10.1016/j.enconman.2016.06.053.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS)*, pages 1657–1665, 2015.

Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* The MIT Press, 2002.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13:1393–1434, 2012.

A. Tascikaraoglu and M. Uzunoglu. A review of combined approaches for prediction of short-term wind speed and power. *Renewable and Sustainable Energy Reviews*, 34:243–254, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13, 2001.

L.J. Wilson and M. Vallée. The Canadian Updateable Model Output Statistics (UMOS) systemm: Design and Development tests. *Wea. Forecast.*, 17:206–222, 2002.

M. Zamo, L. Bel, O. Mestre, and J. Stein. Improved gridded wind speed forecasts by statistical postprocessing of numerical models with block regression. *Weather and Forecasting*, 31:1929–1945, 2016. doi: 10.1175/WAF-D-16-0052.1.

Qinyi Zhang, Sarah Filippi, Arthur Gretton, and Dino Sejdinovic. Large-scale kernel methods for independence testing. *Statistics and Computing*, 28, 1 2018.

Supplementary material (experimental details)

This short supplement is dedicated to the full description of the parameters that we use in the experiments.

A Importance of variables and their evolution through time

A.1 LASSO

The main parameter of the LASSO is the regularization intensity λ. For each data split, we select it based on the NRMSE achieved on the validation set. We consider values in a geometric grid of size 30 ranging from 10^{-5} to 1.

A.2 BAHSIC

We use a Gaussian kernel for both for the input kernel and the output one:

$$k_\gamma(\mathbf{z}, \mathbf{z}') := \exp\left(-\gamma(\|\mathbf{z} - \mathbf{z}'\|_2^2\right).$$

We follow [Song et al., 2012] in the choice of the parameter γ. We standardized both our input and output data so we can apply their heuristic: set this parameter to $\frac{1}{2d}$ where d is the dimension of the inputs of the kernel. Then for the input kernel we have $d = q$ and for the output one $d = m$.

For the Nyström approximation, we use fewer points than for the KRR since as highlighted in [Zhang et al., 2018], for detection of dependency, a fewer number of anchor points are generally sufficient. We then use 100 points for both the input and output approximation.

B Wind speed and wind power forecasting

As a first general note, since we standardized all the variables, we consider the same parameter ranges for prediction of wind speed and wind power.

LASSO The fitted models used for interpretation in the variable selection section are the same that we use here (so the considered parameters are the same).

OLS f-stagewise We selected on the validation set the number of included variables. We consider the following number of variables: $\{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 20\}$.

Nyström KRR We select both the input Gaussian kernel's γ parameter and the regularization parameter λ. We consider the following values:

- γ in a geometric space of length 30 ranging from 10^{-6} to 10^{-3}.
- λ in a geometric space of length 30 ranging from 10^{-4} to 5.

For the Nyström approximation, we use 300 sampled points.

Xg-Boost For Xg-Boost, we validate the trees' maximum depth considering values in $\{3, 4, 5, 6\}$ as well as the minimum loss reduction parameter for values in a geometric space of size 50 ranging from 10^{-7} to 50.

Feedforward NN We consider a NN with 3 hidden layers and validate the number of neurons per layer choosing among the possible values $\{(35, 20, 5), (50, 25, 10), (50, 35, 20), (75, 50, 25)\}$.